## Remarks/Arguments

Claims 1-48 are pending in the current application. Claims 1-24 have been canceled. Applicants reserve the right to file a continuation application for the subject matter contained therein, and have cancelled the claims solely to expedite prosecution of the instant application

**Information Disclosure Statement**

The Examiner states in the Office Action that the Information Disclosure Statements (IDS) filed September 18, 2001 and May 14, 2001 fail to comply with the provisions of 37 CFR §1.97, §1.98 and MPEP §609 because the publication date for De Remer et al. reference was hand written and the U.S. Patent No. 5,555,345 publication data was incorrectly entered. Applicants submit that the publication data for the De Remer et al. reference was printed on the top of the second page of the reference. A copy of the reference is enclosed for Examiner's convenience. Applicants also submit herewith a Supplemental Information Disclosure Statement to correct the publication date of U.S. Patent No. 5,555,345.

**Objection to Declaration**

The Examiner has stated that the Oath/Declaration is defective because the second inventor's mailing address is not legible. In response to this, Applicants hereby submit an Application Data Sheet providing the correct and complete address for the inventor.

**Objection to Specification**

In the above-referenced Office Action, the Examiner objected to the disclosure because of several informalities. More specifically, the Examiner asserts that Block 13 is better represented as step 13 in the amended paragraph of page 17, lines 18-21. The Examiner also asserts that Block 21 is better represented as step 21 on page 13, line 25. The Examiner further asserts that Block 37 is better represented as step 37 on page 20, line 19 and page 21, line 11. The Examiner also asserted that Block 56 is better represented as step 56 on page 23, lines 9-10. Applicants respectfully traverse the Examiner's objections. Applicants assert that the substitution of the term "step" for the term "Block" would lead to a different interpretation of the figures. A linear, fixed progression is typically associated with the term "step", and as is

apparent from blocks 13, 21, 24, 30, 37, 41, 44, 46, and 56 within figures 6-8, 11, 12, 15, and 16, the flow chart illustrated in the figures can include several branches, which results is a nonlinear flow. Thus, Applicants respectfully request that the Examiner withdraw his objection to these aspects of the specification.

The Examiner also objected to the disclosure, asserting that the reason for choosing two identities in Fig. 10B is unclear from the description on page 19, lines 11-14. Applicants respectfully call the Examiner's attention to page 14, line 21, through page 15, line 20, for a description of the logic behind Fig. 10B. Applicants respectfully request that the Examiner withdraw his objection.

**Claim Rejections – 35 U.S.C. §102**

The Examiner rejected Claims 25-48 under 35 U.S.C. §102(b) as being anticipated by Simonyi, U.S. Patent No. 5,911,072. Applicants respectfully traverse the Examiner's rejection.

**Claim 25, 26, and 27**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teach receiving the input from a user as recited in the first element of Applicants' Claim 25. Applicants respectfully submit that column 5, lines 14-17 of the Simonyi '072 reference teaches that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants' invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several

advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 25. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teach assisting the user in constructing a well-architected IP tree. By contrast, the second element of Applicants' Claim 25 is a more powerful process which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches self-activating the source tree to interpret the user input, as recited in the third element of Applicants' Claim 25. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention, as recited in Claim 25, is directed toward a method through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 25, and especially such structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 25.

Claims 26 and 27 depend from Claim 25. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the

independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 26 and 27, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 26 and 27.

**Claims 28-30**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teaches means for receiving the input from a user as recited in the first element of Applicants' Claim 28. Applicants submit that column 5, lines 14-17 teaches that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants' invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches means for matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 28. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teaches means for assisting the user in constructing a well-architected IP tree. By contrast, the second

element of Applicants' Claim 28 is a more powerful system which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches means for self-activating the source tree to interpret the user input, as recited in the third element of Applicants' Claim 28. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention, as recited in Claim 28, is directed toward a system through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 28, and especially such structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 28.

Claims 29 and 30 depend from Claim 28. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 29 and 30, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 29 and 30.

**Claims 31-32**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teach a computer readable program code means for receiving the input from a user as recited in the first element of Applicants' Claim 31. Applicants respectfully submit that column 5, lines 14-17 of the Simonyi '072 reference teaches that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants' invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches computer readable program code means for matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 31. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teaches a computer readable program code means for assisting the user in constructing a well-architected IP tree. By contrast, the second element of Applicants' Claim 31 is a more powerful process which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches computer readable program code means for self-activating the source tree to interpret the

user input, as recited in the third element of Applicants' Claim 31. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention, as recited in Claim 31, is directed toward computer readable program code means through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 31, and especially such structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 31.

Claims 32 and 33 depend from Claim 31. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 32 and 33, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 32 and 33.

**Claim 34-38**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teach a method for receiving the input from a user as recited in the first element of Applicants' Claim 34. Applicants respectfully submit that column 5, lines 14-17 of the Simonyi '072 reference teaches

that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants' invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 34. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teach assisting the user in constructing a well-architected IP tree. By contrast, the second element of Applicants' Claim 34 is a more powerful process which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches self-activating the source tree to interpret the user input, as recited in the third element of Applicants' Claim 34. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention,

as recited in Claim 34, is directed toward a method through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 34, and especially such structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 34.

Claims 35-38 depend from Claim 34. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 35-38, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 35-38.

**Claims 39-43**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teach receiving the input from a user as recited in the first element of Applicants' Claim 39. Applicants respectfully submit that column 5, lines 14-17 of the Simonyi '072 reference teaches that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that

are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants' invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches means for matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 39. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teach assisting the user in constructing a well-architected IP tree. By contrast, the second element of Applicants' Claim 39 is a more powerful system which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches self-activating the source tree to interpret the user input, as recited in the third element of Applicants' Claim 39. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention, as recited in Claim 39, is directed toward a system through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 39, and especially such structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 39.

Claims 40-43 depend from Claim 39. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 40-43, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 40-43.

**Claims 44-48**

The Examiner asserts that column 5, lines 14-17 of the Simonyi '072 reference teach a computer readable program code means for receiving the input from a user as recited in the first element of Applicants' Claim 44. Applicants respectfully submit that column 5, lines 14-17 of the Simonyi '072 reference teaches that users utilize a graphical user interface to directly manipulate an IP tree. At column 5, lines 45-57, the Simonyi '072 reference teaches that an IP tree is a tree data structure that contains nodes, with each node containing an operator field, each operator field in turn containing a reference to a node corresponding to a declaration of an IP computational construct or a program-defined declaration. The operand fields contain references to subtrees of the IP tree that represent operands for the node. Each IP tree includes an IP set of "IP declaration nodes" that are defined by the IP system. An IP declaration node is a node with an operator field that references the "declaration" IP computational construct. From these definitions it is clear that in the Simonyi '072 reference, a user is directly involved in entering and modifying an IP tree for use in a conventional syntax parser. By contrast, Applicants'

invention is designed to process user input as it occurs, without the need for users to interact with a conventional syntax parser. Applicants' invention is also designed to dynamically create one or more trees based on the user input. This makes Applicants' invention language and syntax neutral, which provides several advantages over the Simonyi '072 reference, and distinguishes the claim over the Simonyi '072 reference.

The Examiner asserted that column 24, lines 30-43 of the Simonyi '072 reference teaches means for matching concept representatives to the user input using experimentation to result in a source tree, as recited in the second element of Applicants' Claim 44. Applicants respectfully traverse the Examiner's assertion. Column 24, lines 30-43 of the Simonyi reference teach assisting the user in constructing a well-architected IP tree. By contrast, the second element of Applicants' Claim 44 is a more powerful system which involves automatically converting the syntax-free user input received as part of the first element into a source tree by experimentally matching concept representatives to the user input.

The Examiner also asserted that column 28, lines 38-57 of the Simonyi '072 reference teaches self-activating the source tree to interpret the user input, as recited in the third element of Applicants' Claim 44. Applicants respectfully traverse the Examiner's assertion. Column 28, lines 38-57 describes pseudo-code for the "average" enzyme, which transforms an expression such as "Average (3.4, 2.8, 1.5)" into the simpler, prefix form that uses only binary operators "divide(add(add(3.4, 2.8), 1.5), 3)". This is simply an interpretation of a command written in one language into a command written in another language. By contrast, Applicants' invention, as recited in Claim 44, is directed toward a system through which user input is received without foreknowledge of the structure of the input, concept representatives are applied to the input to determine a proper source tree structure using experimentation, and the resulting source tree is applied to the user input. This concept is neither taught nor suggested by Simonyi.

The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabirk Gmbh v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Simonyi clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent claim 44, and especially such

structure arranged as in the claims. Thus, Applicants' invention is patentable over Simonyi, and Applicants respectfully request that the Examiner withdraw his rejection of Claim 44.

Claims 45-48 depend from Claim 44. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a valid independent claim, the independent claim is *a fortiori* valid because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments above for each of Claims 45-48, and respectfully requests that the Examiner find the claims patentable over Simonyi. In the event the Examiner does not find Applicants' arguments to be persuasive, Applicants reserve the right to file a substantive response to the Examiner's arguments with respect to Claims 45-48.

## CONCLUSION

Having responded to all objections and rejections set forth in the outstanding Office Action, it is submitted that claims 25-48 are in condition for allowance and Notice to that effect is respectfully solicited. In the event that the Examiner is of the opinion that a brief telephone or personal interview will facilitate allowance of one or more of the above claims, he is courteously requested to contact applicant's undersigned representative.

Respectfully submitted,

James E. Goepel (Reg. No. 50,851)
Agent for Applicant

GREENBERG TRAURIG
1750 Tyson's Boulevard
Suite 1200
McLean, VA 22102
(703) 903-7536
E-mail: goepelj@gtlaw.com

Filed: September 17, 2004